**Guidelines for code Documentation**

Documenting the Raptat code base with properly formatted comments and Javadoc annotations is very important for readability and maintenance. These guidelines are developed to maintain consistency in the appearance and usage of comments throughout the application.

***General conventions for comments***:

1. Limit comments to 80 characters per line.
2. Be consistent in type of commenting syntax used. Use /\*\* to start the comment and \*/ to end the comment. Trailing asterisk \*\*\*\* are not required. Single line comments can also use // (double slash).
3. Follow proper indentation conventions such as 4 spaces in Java.
4. Class/interface and method names must be self-descriptive and not too long.
5. The naming convention for classes/interfaces: start the first word of the name with caps and then follow with camelCase for the rest of the words. E.g, *POSIntegrator* class name. All other names such as variable names or method names use camelCase only. E.g, *employeeName* variable *getIdNumber* () method name.
6. The first summary sentence in the comments section of any class/interface or method declaration must be a concise, clear and complete description of its function.
7. The description needs to be implementation independent.
8. Use complete sentences in descriptions, 3rd person declarative. Ex: “returns the list”
9. The summary descriptions must use words other than those in the name of the method/class/interface.
10. Avoid abbreviations such as i.e, a.k.a, viz. Use expanded words like “that is” and “namely” instead.
11. All explicit declarations of constructors must be described with comments.
12. Comment the code in between the constructs ONLY when the logic/flow of the code is not self-explanatory. Keep the comments formatted to the side (preferably) of the code and as short as possible.
13. Do not write comments as placeholders with what you plan to do in a particular method/class /interface, if you have not yet implemented it.
14. Author names must be real and not fake names.

***Javadoc conventions***: (in addition to above)

1. These Class/interface/method annotations must appear in the following order. All tags are not required to be used at all times.

@author---------------------------required for class/interface

@version--------------------------required for class/interface

@constructor--------------------------required for constructors

@param---------------------------required for constructors and methods

@return----------------------------required for methods

@exception /@throws----------required for methods that throw checked exceptions

@see----required for linking other resources (methods/classes/interfaces) for reference.

@since ---required to indicate the version number of the project

@serial (or @serialField or @serialData)-------------required for serialization

@deprecated ----------must be followed by a newline or space

1. While describing @param and @return, use verb phrases such as “tokenizes the sentences”, “gets the parts of speech”. Start the phrase without capitalization and do not end with a period. This convention is followed *only* if the description is one single phrase. If using multiple phrases, use normal sentence conventions with capitalization and full stops.

E.g., *@param String name represents the full name*

*@param String name Represents the full name. Uses first, last convention.*

For any further information on Javadoc comments, please refer to the link below.

**http://download.oracle.com/javase/1.5.0/docs/tooldocs/windows/javadoc.html**